Simple Object Access Protocol (SOAP) 1.1

W3C Note 08 May 2000

This version:
http://www.w3.org/TR/2000/NOTE-SOAP-20000508
Latest version:
http://www.w3.org/TR/SOAP
Authors (alphabetically):

Don Box, DevelopMentor
David Ehnebuske, IBM
Gopal Kakivaya, Microsoft
Andrew Layman, Microsoft
Noah Mendelsohn, Lotus Development Corp.
Henrik Frystyk Nielsen, Microsoft
Satish Thatte, Microsoft
Dave Winer, UserLand Software, Inc.

Abstract

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols; however, the only bindings defined in this document describe how to use SOAP in combination with HTTP and HTTP Extension Framework.

Status

This document is a submission to the World Wide Web Consortium (see Submission Request, W3C Staff Comment) to propose the formation of a working group in the area of XML-based protocols. Comments are welcome to the authors but you are encouraged to share your views on the W3C's public mailing list <xml-dist-app@w3.org> (see archives).

This document is a NOTE made available by the W3C for discussion only. Publication of this Note by W3C indicates no endorsement by W3C or the W3C Team, or any W3C Members. W3C has had no editorial control over the preparation of this Note. This document is a work in progress and may be updated, replaced, or rendered obsolete by other documents at any time.

A list of current W3C technical documents can be found at the Technical Reports page.

2. The SOAP Message Exchange Model

SOAP messages are fundamentally one-way transmissions from a sender to a receiver, but as illustrated above, SOAP messages are often combined to implement patterns such as request/response.

SOAP implementations can be optimized to exploit the unique characteristics of particular network systems. For example, the HTTP binding described in section 6 provides for SOAP response messages to be delivered as HTTP responses, using the same connection as the inbound request.

Regardless of the protocol to which SOAP is bound, messages are routed along a so-called "message path", which allows for processing at one or more intermediate nodes in addition to the ultimate destination.

A SOAP application receiving a SOAP message MUST process that message by performing the following actions in the order listed below:

1. Identify all parts of the SOAP message intended for that application (see section 4.2.2)
2. Verify that all mandatory parts identified in step 1 are supported by the application for this message (see section 4.2.3) and process them accordingly. If this is not the case then discard the message (see section 4.4). The processor MAY ignore optional parts identified in step 1 without affecting the outcome of the processing.
3. If the SOAP application is not the ultimate destination of the message then remove all parts identified in step 1 before forwarding the message.

Processing a message or a part of a message requires that the SOAP processor understands, among other things, the exchange pattern being used (one way, request/response, multicast, etc.), the role of the recipient in that pattern, the employment (if any) of RPC mechanisms such as the one documented in section 7, the representation or encoding of data, as well as other semantics necessary for correct processing.

While attributes such as the SOAP encodingStyle attribute (see section 4.1.1) can be used to describe certain aspects of a message, this specification does not mandate a particular means by which the recipient makes such determinations in general. For example, certain applications will understand that a particular <getStockPrice> element signals an RPC request using the conventions of section 7, while another application may infer that all traffic directed to it is encoded as one way messages.

3. Relation to XML

All SOAP messages are encoded using XML (see [7] for more information on XML).

A SOAP application SHOULD include the proper SOAP namespace on all elements and attributes defined by SOAP in messages that it generates. A SOAP application MUST be able to process SOAP namespaces in messages that it receives. It MUST discard messages that have incorrect namespaces (see section 4.4) and it MAY process SOAP messages without SOAP namespaces as though they had the correct SOAP namespaces.

SOAP defines two namespaces (see [8] for more information on XML namespaces):

- The SOAP envelope has the namespace identifier "http://schemas.xmlsoap.org/soap/envelope/"
- The SOAP serialization has the namespace identifier "http://schemas.xmlsoap.org/soap/encoding/"

A SOAP message MUST NOT contain a Document Type Declaration. A SOAP message MUST NOT contain Processing Instructions. [7]

SOAP uses the local, unqualified "id" attribute of type "ID" to specify the unique identifier of an encoded element. SOAP uses the local, unqualified attribute "href" of type "uri-reference" to specify a reference to that value, in a manner conforming to the XML Specification [7], XML Schema Specification [11], and XML Linking Language Specification [9].

With the exception of the SOAP mustUnderstand attribute (see section 4.2.3) and the SOAP actor attribute (see section 4.2.2), it is generally permissible to have attributes and their values appear in XML instances or alternatively in schemas, with equal effect. That is, declaration in a DTD or schema with a default or fixed value is semantically equivalent to appearance in an instance.

4. SOAP Envelope

A SOAP message is an XML document that consists of a mandatory SOAP envelope, an optional SOAP header, and a mandatory SOAP body. This XML document is referred to as a SOAP message for the rest of this specification. The namespace identifier for the elements and attributes defined in this section is "http://schemas.xmlsoap.org/soap/envelope/". A SOAP message contains the following:

- The Envelope is the top element of the XML document representing the message.
- The Header is a generic mechanism for adding features to a SOAP message in a decentralized manner without prior agreement between the communicating parties. SOAP defines a few attributes that can be used to indicate who should deal with a feature and whether it is optional or mandatory (see section 4.2)
- The Body is a container for mandatory information intended for the ultimate recipient of the message (see section 4.3). SOAP defines one element for the body, which is the Fault element used for reporting errors.

The grammar rules are as follows:

1. Envelope
   - The element name is "Envelope".
   - The element MUST be present in a SOAP message
   - The element MAY contain namespace declarations as well as additional attributes. If present, such additional attributes MUST be namespace-qualified. Similarly, the element MAY contain additional sub elements. If present these elements MUST be namespace-qualified and MUST follow the SOAP Body element.
2. Header (see section 4.2)
   - The element name is "Header".
   - The element MAY be present in a SOAP message. If present, the element MUST be the first immediate child element of a SOAP Envelope element.
   - The element MAY contain a set of header entries each being an immediate child element of the SOAP Header element. All immediate child elements of the SOAP Header element MUST be namespace-qualified.
3. Body (see section 4.3)
   - The element name is "Body".
   - The element MUST be present in a SOAP message and MUST be an immediate child element of a SOAP Envelope element. It MUST directly follow the SOAP Header element if present. Otherwise it MUST be the first immediate child element of the SOAP Envelope element.
   - The element MAY contain a set of body entries each being an immediate child element of the SOAP

XI. Evidence Appendix

Part B